Analysis of Farthest Point Sampling for Approximating Geodesics in a Graph

Pegah Kamousi, Sylvain Lazard, Anil Maheshwari, Stefanie Wuhrer

Analysis of Farthest Point Sampling for Approximating Geodesics in a Graph

Pegah Kamousi[a] Sylvain Lazard[b] Anil Maheshwari[c] Stefanie Wuhrer[d]

April 4, 2016

Abstract

A standard way to approximate the distance between two vertices p and q in a graph is to compute a shortest path from p to q that goes through one of k sources, which are well-chosen vertices. Precomputing the distance between each of the k sources to all vertices yields an efficient computation of approximate distances between any two vertices. One standard method for choosing k sources is the so-called *Farthest Point Sampling* (FPS), which starts with a random vertex as the first source, and iteratively selects the farthest vertex from the already selected sources.

In this paper, we analyze the stretch factor $\mathcal{F}_{\text{FPS}}$ of approximate geodesics computed using FPS, which is the maximum, over all pairs of distinct vertices, of their approximated distance over their geodesic distance in the graph. We show that $\mathcal{F}_{\text{FPS}}$ can be bounded in terms of the minimal value $\mathcal{F}^*$ of the stretch factor obtained using an optimal placement of k sources as $\mathcal{F}_{\text{FPS}} \leqslant 2r_e^2\mathcal{F}^* + 2r_e^2 + 8r_e + 1$, where r_e is the length ratio of longest edge over the shortest edge in the graph. We further show that the factor r_e is not an artefact of the analysis by providing a class of graphs for which $\mathcal{F}_{\text{FPS}} \geqslant \frac{1}{2}r_e\mathcal{F}^*$.

Keywords: Farthest Point Sampling; Approximate Geodesics; Shortest Paths; Planar Graphs; Approximation Algorithms

1 Introduction

In the context of shape analysis, it is commonly required to compute and analyze geodesics between many pairs of vertices on a shape that is represented by a connected undirected graph G with n vertices and m edges. To compute shortest-path queries from a single-source on G, Dijkstra's algorithm [4] takes $O(n \log n + m)$ time. To compute

[a]Département d'Informatique, Université Libre de Bruxelles, Belgium, E-mail: pegahk@gmail.com
[b]Inria, Loria, Nancy, France, E-mail: sylvain.lazard@inria.fr
[c]School of Computer Science, Carleton University, Canada, E-mail: anil@scs.carleton.ca
[d]Inria Grenoble Rhône-Alpes, France, E-mail: stefanie.wuhrer@inria.fr

all-pairs shortest-paths, we can run Dijkstra's algorithm starting from each of the n vertices and, while there are more efficient methods, this problem has a trivial $\Omega(n^2)$ lower bound.

To reduce this complexity, the problem of efficiently *approximating* the distance between any two vertices is often considered. A very recent method efficiently computes a $(1+\epsilon)$-approximation to a single such query in a planar graph in $O((\log\log n)^3/\epsilon^2 + \log\log\sqrt{\log\log((\log\log n)/\epsilon^2)}/\epsilon^2)$ time and $O(n((\log\log n)^2/\epsilon + (\log\log n)/\epsilon^2)))$ space [17].

In contrast to this work that builds a carefully chosen data structure, we are interested in a class of simple algorithms, commonly used in practice, that compute in a preprocessing phase a set $S = \{s_1, \ldots, s_k\}$ of k vertices, called *sources*, in G and runs Dijkstra's algorithm from each of them in $O(k\,(n\log n + m))$ total time. Then, the distance between any two vertices p and q is approximated as the minimum, over all k sources, of the distance from p to q through one of the sources. The quality of the worst approximation is characterized by the stretch factor, defined as the maximum, over all pairs of distinct vertices, of their approximated distance over their geodesic distance in the graph, that is

$$\mathcal{F} = \max_{(p,q)\in V,\ p\neq q}\ \min_{s_i\in S} \frac{d(p,s_i)+d(s_i,q)}{d(p,q)},$$

where V denotes the set of vertices of G and where the function $d(.,.)$ measures the shortest geodesic distance between two vertices. Throughout this paper, we use for simplicity the notation $\max_{p,q}$ for $\max_{(p,q)\in V,\ p\neq q}$.

A natural problem is thus to compute an optimal placement of k sources that yields a minimum stretch factor, denoted $\mathcal{F}^*$. We refer to this problem as the *k-center path-dilation problem*. This problem is NP-complete even for planar graphs because the existence of at most k sources so that the stretch factor is 1 is trivially equivalent to the existence of a vertex cover of size at most k, which is NP-complete even for planar graphs [7]. [1] Furthermore, we show in [10] that the k-center path-dilation problem is also NP-complete in the case of planar triangle graphs (i.e., connected graphs whose faces have three edges and whose edges are incident to at most two faces), which are of particular interest for shape analysis.

For computing a set of at most k sources, Könemann et al. [12, Thm. 3] present a simple algorithm that yields a stretch factor $\mathcal{F}_{\mathrm{K}} \leqslant 2\mathcal{F}^* + 1 + \epsilon$ in time $O(k\,(n\log n + m)\log(n\,r_e/\epsilon))$ for any $\epsilon > 0$, where r_e is the lengths ratio of the longest over the shortest edges in G. For convenience of the reader, we detail in Section 2 their algorithm and proof because they missed the ϵ term and did not state the complexity.

In this work, we analyze the stretch factor of the even simpler and commonly used *Farthest Point Sampling* (FPS) heuristic for selecting a set of k sources [9, 14]. FPS starts by selecting a random vertex and iteratively selects a vertex that has the largest geodesic distance to its closest already selected source, until k sources are picked. Running Dijkstra's

[1] A NP-hardness proof for general graphs can also be found in [11, §2] but, as it comes as a corollary of another NP-hardness reduction, it is less trivial and it inherently uses non-planar graphs.

algorithm from each of the sources directly yields a total running in $O(k\,(n\log n + m))$.

To the best of our knowledge, no theoretical results are known on the quality of the stretch factor, $\mathcal{F}_{\mathrm{FPS}}$, obtained by an FPS of k sources, compared to the minimal stretch factor $\mathcal{F}^*$. In this paper, we prove that for any connected undirected graph and any choice of k sources obtained by the FPS algorithm,

$$\mathcal{F}_{\mathrm{FPS}} \leqslant 2r_e^2(\mathcal{F}^* + 1) + 8r_e + 1,$$

where r_e is the lengths ratio of the longest over the shortest edges in G (Theorem 1). We further show that the factor r_e is not an artefact of the analysis by providing a family of graphs for which $\mathcal{F}_{\mathrm{FPS}} \geqslant \frac{1}{2} r_e \mathcal{F}^*$ (Theorem 6). This shows that if the ratio r_e is large, $\mathcal{F}_{\mathrm{FPS}}$ can be much larger than the optimal stretch factor $\mathcal{F}^*$ but, on the other hand, $\mathcal{F}^*$ is likely to be large as well. Indeed, consider a graph with $k+1$ arbitrarily small edges such that all their adjacent edges are long enough: at least one of these edges is not incident to a source and for the endpoints of this edge, their approximated distance over their geodesic distance is arbitrarily large. Note that all our bounds also hold in the case where the edge lengths ratio r_e is defined only by pairs of edges that belong to one and the same shortest path in G.

The relevance of our bounds on $\mathcal{F}_{\mathrm{FPS}}$ is to give some theoretical insight on why FPS has been used successfully in heuristics for shape processing. However, it should be stressed that the edge lengths ratio r_e appears in the upper and lower bounds on $\mathcal{F}_{\mathrm{FPS}}$ but not on $\mathcal{F}_{\mathrm{K}}$. Still, r_e appears in the running time for the latter and not for the former, but since it appears in logarithmic form, it is fair to expect that Könemann et al.'s algorithm would give better results in terms of the combination of stretch factors and running times than the widely used FPS algorithm. Nevertheless, we are not aware of any experimental study on the subject.

After discussing related work in Section 2, we prove our main results, Theorems 1 and 6 in Sections 3.1 and 3.2 respectively.

2 Related Work

Computing geodesics on polyhedral surfaces is a well-studied problem for which we refer to the recent survey by Bose et al. [2]. While much work on surface processing compute geodesics that are allowed to pass through the interior of faces, much work also restrict geodesics to go through vertices and edges, as they are easy to compute. In this paper, we restrict geodesics to be shortest paths along edges of the underlying graph.

The FPS algorithm has been used for a variety of surface processing tasks. The algorithm was first introduced for graph clustering [9], and later independently developed for 2D images [6] and extended to 3D meshes [14]. This sampling strategy has been used to efficiently compute approximate geodesic distances [1, 8], to recognize the class of an input shape [5, 13], and to compute point-to-point correspondences between surfaces [3, 16, 15]. In practice, $\mathcal{F}_{\mathrm{FPS}}$ and r_e are typically reasonably small. For instance, for the Greek head model consisting of 6607 vertices used by Ruggeri and Saupe [15], it is $r_e = 15.4$ and $\mathcal{F}_{\text{FPS}} = 18.8$ for $k = 500$. In our implementation, we computed $\mathcal{F}_{\text{FPS}}$ by averaging over five sets of randomly chosen sources computed using FPS.

The problem we study is closely related to the k-center problem, which aims at finding k centers (or sources) $S' = \{s'_1, \ldots, s'_k\}$, such that the maximum distance of any point to its closest center is minimized. With the notation defined above, the k-center problem aims at finding S', such that $\max_p (\min_i d(p, s'_i))$ is minimized. This problem is NP-hard and FPS gives a 2-approximation, which means that k centers $S = \{s_1, \ldots, s_k\}$ found using FPS have the property that $\max_p (\min_i d(p, s_i)) \leqslant 2 \max_p (\min_i d(p, s'_i))$ [9].

In the context of isometry-invariant shape processing, we are interested in bounding the stretch induced by the approximation rather than ensuring that every point has a close-by source. A related problem that has been studied in the context of networks by Könemann et al. [12] is the *edge-dilation k-center problem*. Here, every point p is assigned a source, denoted by s_p, and the distance between two points p and q is approximated by the length of the shortest path through p, s_p, s_q, and q. The aim is then to find a set of sources that minimizes the worst stretch, and Könemann et al. show that this problem is NP-hard and propose an approximation algorithm to solve the problem.

Könemann et al. [12, Theorem 3] also study a modified version of the above problem, which is similar to our problem. In particular and as mentioned in Section 1, they present an algorithm for computing at most k sources that yields a stretch factor of at most $2\mathcal{F}^* + 1 + \epsilon$ in time $O(k\,(n \log n + m) \log(n\, r_e/\epsilon))$ for any $\epsilon > 0$, where r_e is the lengths ratio of the longest over the shortest edges in G. Their algorithm and proof go as follows.

For any value α, their basic routine iteratively includes in a set $S(\alpha)$ an endpoint of the shortest edge that cannot yet be approximated with a stretch factor $2\alpha + 1$, until no such edges are left or $|S(\alpha)| > k$ [12, §3.3].[2] Then, denoting $\operatorname{diam}(G)$ the diameter of the graph and $\ell_{\min}$ the length of its shortest edge, they perform a binary search in $[1, 2\operatorname{diam}(G)/\ell_{\min}]$ to obtain an interval $[a, b]$ of length at most $\epsilon/2$ so that $|S(a)| > k$ and $|S(b)| \leqslant k$, and they output $S(b)$.[3] They prove that $|S(\mathcal{F}^*)| \leqslant k_{\mathcal{F}^*} \leqslant k$, where $k_{\mathcal{F}^*}$ is the minimum number of sites that can realize the stretch factor $\mathcal{F}^*$ [12, Lemmas 1 and 3]. Furthermore, since $|S(\alpha)|$ is non-increasing,[4] $a < \mathcal{F}^*$ and thus $b < \mathcal{F}^* + \epsilon/2$. Hence, $S(b)$ is a set of at most k sites that realize a stretch factor of at most $2\,\mathcal{F}^* + 1 + \epsilon$. Finally, the above complexity is straightforward since every call to the basic routine essentially amounts to $O(k)$ calls to Dijkstra's algorithm and the binary search performs $O(\log(\frac{\operatorname{diam}(G)}{\ell_{\min}\epsilon}))$, which is also $O(\log(n\, r_e/\epsilon))$, calls to the basic routine.

[2] If $|S(\alpha)| \leqslant k$, the fact that $S(\alpha)$ induces a stretch factor of at most $2\alpha + 1$ is not discussed in [12] but it follows from Lemma 2 below.

[3] The binary search is stated without stopping criteria in [12, Proof of Lemma 2], which is why the ϵ term is missing from their approximation factor.

[4] Although not mentioned in [12], this follows from the same arguments used in the proof of [12, Lemma 3].

3 Approximating Geodesics with Farthest Point Sampling

We start this section with some definitions and notation. We consider a connected graph G in which the edges have lengths from a positive and finite interval $[\ell_{min}, \ell_{max}]$, and r_e denotes the ratio ℓ_{max}/ℓ_{min}. We require the graph to be connected so that the distance between any two vertices is finite. In this section, we do not require the graph to satisfy any other criteria, but observe that if it is not planar, the running time of FPS will be $O(k(m + n \log n))$, where m is the number of edges.

Given k vertices (sources) $S = \{s_1, \ldots, s_k\}$ in the graph, let s_p denote the (or a) closest source to a vertex p and let $d(p, S, q)$ denote the shortest path length from p to q through any source in S, that is $\min_i(d(p, s_i) + d(s_i, q))$. Let $S^* = \{s_1^*, \ldots, s_k^*\}$ be a choice of sources that minimizes the stretch factor $\mathcal{F}^* = \max_{p,q} d(p, S^*, q)/d(p, q)$. Furthermore, let $S' = \{s_1', \ldots, s_k'\}$ be a choice of sources that minimizes $\max_p d(p, s_p)$. In other words, the set S^* is an optimal solution to k-center path-dilation problem and the set S' is an optimal solution to the k-center problem.

This section provides upper and lower bounds for $\mathcal{F}_{\text{FPS}}$.

3.1 Upper Bound

In this section, we prove the following theorem.

Theorem 1. *For any set of k sources returned by the FPS algorithm on a connected graph G with edge lengths of ratio at most r_e, $\mathcal{F}_{FPS} \leqslant 2r_e^2(\mathcal{F}^* + 1) + 8r_e + 1$.*

Let $S = \{s_1, \ldots, s_k\}$ be a set of sources returned by the FPS algorithm on a connected graph G. In order to prove this theorem, we first show that, for any set of sources, the stretch factor $\max_{p,q} \frac{d(p,S,q)}{d(p,q)}$ is realized when p and q are adjacent in the graph (Lemma 2). We use this property to bound this stretch factor in terms of $\max_p d(p, s_p)$ (Lemma 3). On the other hand, we bound the stretch factor of any set of sites in terms of the stretch factor of an optimal set of sources for the k-center problem (Lemma 5). We then combine these results to prove Theorem 1.

Lemma 2. *For any sources $S = \{s_1, \ldots, s_k\}$ and any given vertex q in G, the maximum ratio $\max_p \frac{d(p,S,q)}{d(p,q)}$ is realized for some p that is adjacent to q in G. It follows that the maximum ratio $\max_{p,q} \frac{d(p,S,q)}{d(p,q)}$ is realized for some p and q that are adjacent in G.*

Proof. For the sake of contradiction, let q be any fixed vertex and let p be a non-adjacent vertex that realizes the maximum $\max_p \frac{d(p,S,q)}{d(p,q)}$ and such that among all the vertices p' that realize this maximum, the shortest path from p to q has the smallest number of edges.

Let $\tilde{p}$ be the immediate neighbor of p along the shortest path from p to q. As before, $d(\tilde{p}, S, q)$ denotes the shortest path length from $\tilde{p}$ to q through any source in S. Let ℓ be the length of the edge $p\tilde{p}$ (see Figure 1). We have $d(\tilde{p}, S, q) \geqslant d(p, S, q) - \ell$. Dividing by $d(\tilde{p}, q) = d(p, q) - \ell$ we get

$$\frac{d(\tilde{p}, S, q)}{d(\tilde{p}, q)} \geqslant \frac{d(p, S, q)}{d(p, q) - \ell} - \frac{\ell}{d(\tilde{p}, q)}.$$

On the other hand, by multiplying $d(p, q) = d(\tilde{p}, q) + \ell$ by $\frac{d(p,S,q)}{d(p,q)d(\tilde{p},q)}$ we have

$$\frac{d(p, S, q)}{d(p, q) - \ell} = \frac{d(p, S, q)}{d(p, q)} + \frac{\ell d(p, S, q)}{d(\tilde{p}, q) \cdot d(p, q)},$$

and therefore

$$\frac{d(\tilde{p}, S, q)}{d(\tilde{p}, q)} \geqslant \frac{d(p, S, q)}{d(p, q)} + \frac{\ell}{d(\tilde{p}, q)} \cdot \left(\frac{d(p, S, q)}{d(p, q)} - 1\right) \geqslant \frac{d(p, S, q)}{d(p, q)},$$

which contradicts our assumption. Indeed, either the inequality is strict and $\frac{d(p,S,q)}{d(p,q)}$ was not maximum, or the equality holds and the shortest path from $\tilde{p}$ to q has fewer edges than the shortest path from p to q. □

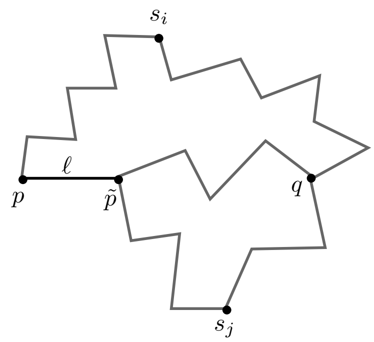

Figure 1: For the proof of Lemma 2.

The property of the previous lemma that $\max_{p,q} \frac{d(p,S,q)}{d(p,q)}$ is realized when p and q are neighbors allows us to bound it as follows.

Lemma 3. *For any sources $S = \{s_1, \ldots, s_k\}$, we have*

$$\frac{2}{\ell_{max}} \max_p d(p, s_p) - 1 \leqslant \max_{p,q} \frac{d(p, S, q)}{d(p, q)} \leqslant \frac{2}{\ell_{min}} \max_p d(p, s_p) + 1.$$

Proof. For the upper bound, we have $d(p, S, q) \leqslant d(p, s_p) + d(s_p, q) \leqslant 2d(p, s_p) + d(p, q)$. Therefore, $\frac{d(p,S,q)}{d(p,q)} \leqslant \frac{2}{d(p,q)} d(p, s_p) + 1$. This holds for any vertices p and q and thus for those that realize the maximum of $\frac{d(p,S,q)}{d(p,q)}$. Furthermore, $d(p, q) \geqslant \ell_{min}$ and $d(p, s_p) \leqslant \max_p d(p, s_p)$. Hence,

$$\max_{p,q} \frac{d(p, S, q)}{d(p, q)} \leqslant \frac{2}{\ell_{min}} \max_p d(p, s_p) + 1.$$

For the lower bound, we have by the triangle inequality that, for any i, $d(q, s_i) \geqslant d(p, s_i) - d(p, q)$. Adding $d(p, s_i)$ on both sides, we get $d(p, s_i) + d(q, s_i) \geqslant 2d(p, s_i) - d(p, q)$. By the definition of s_p, $d(p, s_i) \geqslant d(p, s_p)$ for any i, thus $d(p, s_i) + d(q, s_i) \geqslant 2d(p, s_p) - d(p, q)$. This holds for any i and thus for the i such that $d(p, s_i) + d(q, s_i)$ is minimum, hence $d(p, S, q) \geqslant 2d(p, s_p) - d(p, q)$. Dividing by $d(p, q)$, we get $\frac{d(p,S,q)}{d(p,q)} \geqslant \frac{2}{d(p,q)} d(p, s_p) - 1$. This holds for any p and q and thus for the vertex p that realizes the maximum of $d(p, s_p)$; let $\bar{p}$ denote such a vertex. We then have that $\frac{d(\bar{p},S,q)}{d(\bar{p},q)} \geqslant \frac{2}{d(\bar{p},q)} \max_p d(p, s_p) - 1$. This holds for any q and in particular for the one that realizes $\max_q \frac{d(\bar{p},S,q)}{d(\bar{p},q)}$. By Lemma 2, the maximum is realized for a q that is adjacent to $\bar{p}$ in G, thus, for such a q, $\frac{2}{d(\bar{p},q)} \geqslant \frac{2}{\ell_{max}}$. It follows that

$$\max_{p,q} \frac{d(p, S, q)}{d(p, q)} \geqslant \max_{q} \frac{d(\bar{p}, S, q)}{d(\bar{p}, q)} \geqslant \frac{2}{\ell_{max}} \max_{p} d(p, s_p) - 1.$$

□

The following lemma bounds the path length between two vertices u and v passing through s_u in terms of the shortest path between u and v through any source.

Lemma 4. *For any sources $S = \{s_1, \ldots, s_k\}$, and vertices u, v we have*

$$d(u, s_u) + d(s_u, v) \leqslant d(u, S, v) + 2d(u, v).$$

Proof. Denote by s_i the source that realizes the minimum $d(u, S, v) = \min_i(d(u, s_i) + d(s_i, v))$. Since by definition $d(u, s_u) \leqslant d(u, s_i)$, we only have to show that $d(v, s_u) \leqslant d(s_i, v) + 2d(u, v)$. Using the triangle inequality twice, we have

$$d(v, s_u) \leqslant d(v, u) + d(u, s_u) \leqslant d(v, u) + d(u, s_i) \leqslant d(v, u) + d(u, v) + d(v, s_i),$$

which concludes the proof. □

These results allow us to bound the stretch factor corresponding to the sources returned by the FPS algorithm with respect to the stretch factor corresponding to an optimal choice of sources for the k-center problem.

Lemma 5. *Let $S = \{s_1, \ldots, s_k\}$ be a set of sources returned by the FPS algorithm and $S' = \{s'_1, \ldots, s'_k\}$ be an optimal set of sources for the k-center problem. Then*

$$\max_{p,q} \frac{d(p, S, q)}{d(p, q)} \leqslant 2r_e \max_{u,v} \frac{d(u, S', v)}{d(u, v)} + 6r_e + 1.$$

Proof. Since S is a set of sources returned by the FPS algorithm, this choice of sources provides a 2-approximation for the k-center problem compared to an optimal solution S'; in other words, $\max_p d(p, s_p) \leqslant 2 \max_p d(p, s'_p)$ [9].

By definition, $d(p, S, q)$ is the minimum over all (fixed) sources s_i of $d(p, s_i) + d(s_i, q)$. Thus, $d(p, S, q) \leqslant d(p, s_p) + d(s_p, q)$. Moreover, by the triangle inequality, $d(s_p, q) \leqslant d(s_p, p) + d(p, q)$ thus $d(p, S, q) \leqslant 2d(p, s_p) + d(p, q)$. One the other hand, $d(p, s_p) \leqslant$

$\max_u d(u, s_u)$, which is less than or equal to $2\max_u d(u, s'_u)$ by the 2-approximation property. For clarity, denote by u the vertex that realizes the maximum $\max_u d(u, s'_u)$. We then have $d(p, S, q) \leqslant 4d(u, s'_u) + d(p, q)$.

Now, by the triangle inequality, $d(u, s'_u) \leqslant d(u, v) + d(v, s'_u)$ for any vertex v. Thus $2d(u, s'_u) \leqslant d(u, v) + d(v, s'_u) + d(u, s'_u)$ which implies, by Lemma 4, that $2d(u, s'_u) \leqslant 3d(u, v) + d(u, S', v)$. Thus, $d(p, S, q) \leqslant 2d(u, S', v) + 6d(u, v) + d(p, q)$ and

$$\frac{d(p, S, q)}{d(p, q)} \leqslant 2\frac{d(u, v)}{d(p, q)}\frac{d(u, S', v)}{d(u, v)} + 6\frac{d(u, v)}{d(p, q)} + 1.$$

This inequality holds for any distinct p and q, and any v distinct from u (recall that u is fixed). Thus it holds for the vertices p and q that realize $\max_{p,q} \frac{d(p,S,q)}{d(p,q)}$ and for the v that realizes $\max_v \frac{d(u,S',v)}{d(u,v)}$. Such a v is a neighbor of u by Lemma 2, thus it satisfies $d(u, v) \leqslant \ell_{max}$. Since $d(p, q) \geqslant \ell_{min}$ for any distinct p and q, and $\max_v \frac{d(u,S',v)}{d(u,v)} \leqslant \max_{u,v} \frac{d(u,S',v)}{d(u,v)}$, we get

$$\max_{p,q} \frac{d(p, S, q)}{d(p, q)} \leqslant 2\frac{\ell_{max}}{\ell_{min}} \max_{u,v} \frac{d(u, S', v)}{d(u, v)} + 6\frac{\ell_{max}}{\ell_{min}} + 1.$$

□

This finally allows us to prove the main theorem.

Proof of Theorem 1. By Lemma 5 and using the same notation, we have

$$\max_{p,q} \frac{d(p, S, q)}{d(p, q)} \leqslant 2r_e \max_{u,v} \frac{d(u, S', v)}{d(u, v)} + 6r_e + 1.$$

Using the upper bound in Lemma 3 on $\max_{u,v} \frac{d(u,S',v)}{d(u,v)}$, we have

$$\max_{p,q} \frac{d(p, S, q)}{d(p, q)} \leqslant 2r_e \left(\frac{2}{\ell_{min}} \max_p d(p, s'_p) + 1\right) + 6r_e + 1.$$

By definition, S' is an optimal set of sources for the k-center problem, that is $\text{argmin}_{s_1,\ldots,s_k} \max_p d(p, s_p)$ and thus $\min_{s_1,\ldots,s_k} \max_p d(p, s_p) = \max_p d(p, s'_p) \leqslant \max_p d(p, s^*_p)$.

We now apply the lower bound of Lemma 3 to $S^* = \{s^*_1, \ldots, s^*_k\}$ which gives

$$\frac{2}{\ell_{max}} \max_p d(p, s^*_p) - 1 \leqslant \max_{p,q} \frac{d(p, S^*, q)}{d(p, q)}$$

and thus

$$\begin{aligned}\max_{p,q} \frac{d(p, S, q)}{d(p, q)} &\leqslant 2r_e \left(\frac{\ell_{max}}{\ell_{min}}(\max_{p,q} \frac{d(p, S^*, q)}{d(p, q)} + 1) + 1\right) + 6r_e + 1\\ &\leqslant 2r_e^2 \max_{p,q} \frac{d(p, S^*, q)}{d(p, q)} + 2r_e^2 + 8r_e + 1.\end{aligned}$$

□

3.2 Lower Bound

We prove in this section the following theorem.

Theorem 6. *For any $n \geqslant 3k$, there exists a connected graph on n vertices with edge lengths of ratio at most r_e such that, for any set of k sources returned by the FPS algorithm, $\mathcal{F}_{FPS} \geqslant \frac{1}{2} r_e \mathcal{F}^*$.*

Consider the graph $\mathcal{G}$ shown in Figure 2 that consists of two subgraphs $\mathcal{C}$ and $\mathcal{D}$ that share a vertex p_1. $\mathcal{C}$ is a chain of $2k$ vertices with edges of length $\ell_{\min} = 1$. $\mathcal{D}$ is a fan of at least k edges connected to p_1, each of length $\ell_{\max} = 2k$. Theorem 6 is a direct consequence of the two following lemmas.

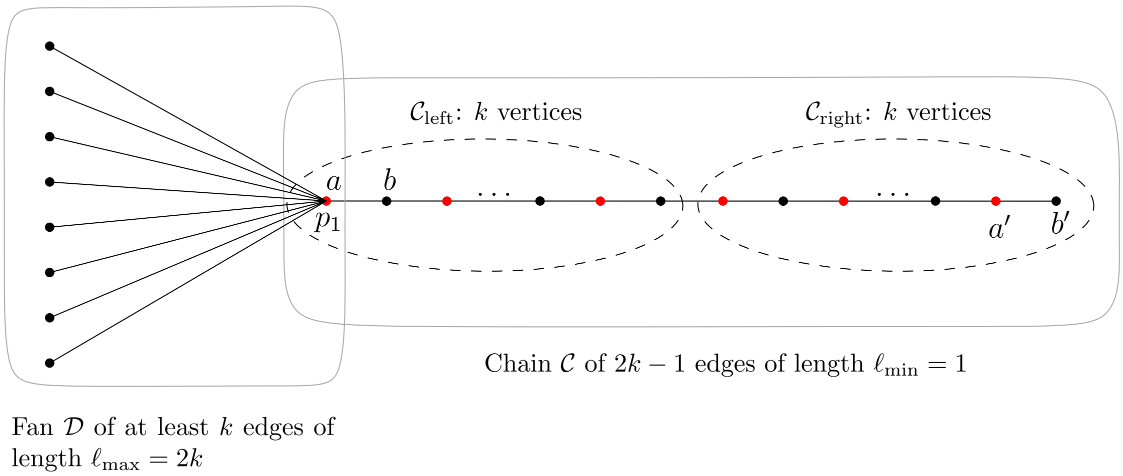

Figure 2: Graph $\mathcal{G}$ for which $\mathcal{F}_{\text{FPS}} \geqslant \frac{1}{2} r_e \mathcal{F}^*$.

Lemma 7. *For any set $S = \{s_1, \ldots, s_k\}$ of sources returned by the FPS algorithm on $\mathcal{G}$,*

$$\max_{p,q} \frac{d(p, S, q)}{d(p, q)} \geqslant r_e/2.$$

Proof. We first show that, among any k sources returned by the FPS algorithm on $\mathcal{G}$, at most one is in $\mathcal{C}$. We can trivially assume that $k > 1$. Let S' denote a set of $k' < k$ sources computed at some point during the FPS algorithm and assume that S' contains exactly one source $\tilde{s}$ in $\mathcal{C}$. Then, any new source is chosen in $\mathcal{D} \setminus \{p_1\}$ because

$$\max_{v \in \mathcal{D} \setminus \{p_1\}} d(v, S') \geqslant \ell_{\max} \quad \text{and} \quad \max_{v \in \mathcal{C}} d(v, \tilde{s}) \leqslant 2k - 1 < \ell_{\max}.$$

Now, as shown in Figure 2, consider in $\mathcal{C}$ two distinct vertices a and b that are the closest to p_1 and two distinct vertices a' and b' that are the farthest from p_1. If no sources of S are in $\mathcal{C}$, then $d(a, S, b) = 2\ell_{\max} + 1$. Otherwise, there is exactly one source in $\mathcal{C}$, say $\tilde{s}$. Let $\mathcal{C}_{\text{left}}$ (resp., $\mathcal{C}_{\text{right}}$) be the subgraph of $\mathcal{C}$ that consists of the $k-1$ edges (and their incident vertices) that are the closest (resp., farthest) to p_1. If $\tilde{s}$ is in $\mathcal{C}_{\text{left}}$, then $d(a', S, b') \geqslant 2k - 1 \geqslant k = \ell_{\max}/2$ and if $\tilde{s}$ is in $\mathcal{C}_{\text{right}}$, then $d(a, S, b) \geqslant \min(2\ell_{\max} + 1, 2k - 1) \geqslant \ell_{\max}/2$. We thus have $\max_{p,q} \frac{d(p,S,q)}{d(p,q)} \geqslant \frac{\ell_{\max}}{2\ell_{\min}}$ since $d(a, b) = d(a', b') = 1 = \ell_{\min}$. $\square$

Lemma 8. *An optimal placement of k sources in $\mathcal{G}$ yields a minimal stretch factor $\mathcal{F}^* = 1$.*

Proof. Consider k specific sources (in red in Figure 2), one of every two vertices in $\mathcal{C}$ starting from p_1. For any two distinct vertices p and q in $\mathcal{G}$, the shortest path from p to q goes through at least one source and thus $\max_{p,q} \frac{d(p,S,q)}{d(p,q)} = 1$. Hence, there exists a choice of k sources so that the stretch factor is 1 and so $\mathcal{F}^* = 1$. □

Acknowledgments

We thank David Mount for suggesting the addition of a lower bound and for providing an initial construction. This research was initiated at Bellairs Workshop on Geometry and Graphs, March 10–15, 2013. We are grateful to Prosenjit Bose, Vida Dujmovic, Stefan Langerman, and Pat Morin for organizing the workshop and to the other workshop participants for providing a stimulating working environment.

References

[1] Z. Ben Azouz, P. Bose, C. Shu, and S. Wuhrer. Approximations of geodesic distances for incomplete triangular manifolds. In *Canadian Conference on Computational Geometry*, 2007.

[2] P. Bose, A. Maheshwari, C. Shu, and S. Wuhrer. A survey of geodesic paths on 3d surfaces. *Computational Geometry - Theory and Applications*, 44(9):486–498, 2011.

[3] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. *Proceedings of the National Academies of Sciences*, 103(5):1168–1172, 2006.

[4] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.

[5] A. Elad and R. Kimmel. On bending invariant signatures for surfaces. *Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1285–1295, 2003.

[6] Y. Eldar, M. Lindenbaum, M. Porat, and Y. Y. Zeevi. The farthest point strategy for progressive image sampling. *Transactions on Image Processing*, 6(9):1305–1315, 1997.

[7] M. R. Garey and D. S. Johnson. The Rectilinear Steiner Tree Problem is NP-Complete. *SIAM Journal on Applied Mathematics*, 32(4):pp. 826–834, 1977.

[8] J. Giard and B. Macq. From mesh parameterization to geodesic distance estimation. In *European Workshop on Computational Geometry*, 2009.

[9] T. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985.

[10] P. Kamousi, S. Lazard, A. Maheshwari, and S. Wuhrer. Analysis of Farthest Point Sampling for Approximating Geodesics in a Graph. Research report, INRIA, November 2013.

[11] J. Könemann, Y. Li, O. Parekh, and A. Sinha. Approximation algorithms for edge-dilation k-center problems. In *Scandinavian Workshop on Algorithms Theory*, 2002.

[12] J. Könemann, Y. Li, A. Sinha, and O. Parekh. An approximation algorithm for the edge-dilation k-center problem. *Operations Research Letters*, 32(5):491–495, 2004.

[13] F. Mémoli and G. Sapiro. Comparing point clouds. In *Symposium on Geometry Processing*, pages 32–40, 2004.

[14] C. Moenning and N. A. Dodgson. Fast marching farthest point sampling. In *Eurographics Poster Presentation*, 2003. Technical Report 562, University of Cambridge Computer Laboratory.

[15] M. Ruggeri and D. Saupe. Isometry-invariant matching of point set surfaces. In *Eurographics Workshop on 3D Object Retrieval*, pages 17–24, 2008.

[16] S. Wuhrer, C. Shu, Z. Ben Azouz, and P. Bose. Posture invariant correspondence of incomplete triangular manifolds. *International Journal of Shape Modeling*, 13(2):139–157, 2007.

[17] C. Wulff-Nielsen. Approximate distance oracles for planar graphs with improved query time-space tradeoff. In *Symposium on Discrete Algorithms*, 2016.